

I cannot conceive that anybody will require multiplications at the rate of 40000 or even 4000 per hour; such a revolutionary change as the octonary scale should not be imposed upon mankind in general for the sake of a few individuals.

F. H. WALES (1936)

VI

DESCRIPCION DEL SISTEMA

CAPITULO VI
DESCRIPCION DEL SISTEMA

VI.1 GENERALIDADES

A partir de los planteamientos expuestos en los dos capítulos precedentes, se ha construido un sistema de Diseño Asistido Optimo por Ordenador denominado DAO².

El sistema DAO² ha sido desarrollado en principio para su aplicación al diseño óptimo estructural, si bien puede ser complementado de forma que su rango de aplicabilidad se extienda a otros problemas de diseño en los que se encuentren subyacentes principios físicos similares a los del cálculo estructural.

En la actualidad el sistema contempla la optimización de estructuras en régimen elástico lineal y estático mediante el Método de Elementos Finitos, abarcando las teorías de:

- Tensión plana.
- Deformación plana.
- Simetría de revolución con cargas axisimétricas.
- Elasticidad tridimensional.

El sistema permite la resolución de los problemas citados y la realización de un análisis de sensibilidad teóricamente exacto hasta el segundo orden respecto a todas las variables de entrada al módulo de cálculo. Las variables de diseño, por consiguiente, pueden afectar tanto a las dimensiones como a la forma de la estructura y a las solicitaciones externas.

DAO², de hecho, no es un programa sino una librería de rutinas escritas en FORTRAN cuya extensión es en la actualidad de aproximadamente 15000 líneas. El diseñador usuario del sistema debe configurar un programa principal, para cada problema de optimización estructural. Este programa o módulo de control, debe realizarse de acuerdo con un patrón preestablecido para obtener la máxima eficiencia, aunque el usuario del sistema goza de una gran flexibilidad al respecto. Para cada problema en particular es necesario generar los correspondientes módulos de decisión y parametrización.

Desde un punto de vista informático hemos de resaltar los siguientes aspectos:

- . La concepción del sistema es totalmente modular.
- . La comunicación entre rutinas se realiza íntegramente mediante transferencia por argumentos.

La concepción modular del sistema permitirá la inclusión en el mismo de otras teorías de cálculo estructural (láminas, problemas no lineales, cálculo matricial de estructuras de barras) y problemas afines, y ampliar la biblioteca de elementos actualmente existente sin alterar la estructura del sistema. Es

posible, igualmente, complementar el sistema con diversos algoritmos de resolución de sistemas de ecuaciones lineales y no lineales, así como con otros métodos de programación matemática.

La transferencia por argumentos permite el dimensionamiento dinámico del programa principal y el acceso del módulo de control a la totalidad de la información.

La versatilidad del sistema permite configurar, dentro de un esquema unificado, diferentes programas de optimización. El usuario del sistema puede configurar su programa de optimización eligiendo entre diversas teorías de cálculo (que pueden trabajar simultáneamente), algoritmos de programación matemática, métodos de resolución de los sistemas de ecuaciones, etc.

VI.2 CARACTERISTICAS DEL SISTEMA

En su estado actual DAO² contempla el cálculo por elementos finitos de estructuras en tensión plana, deformación plana, simetría de revolución con cargas axisimétricas, y elasticidad tridimensional.

La biblioteca de elementos finitos contiene la definición de elementos isoparamétricos de 4, 8 y 9 nodos para problemas planos, y 8, 20 y 27 nodos para problemas tridimensionales.

La integración en el recinto de los elementos se efectúa mediante cuadraturas de Gauss de orden prefijado por el usuario.

El sistema puede tratar varias hipótesis de carga definidas por combinación lineal de los siguientes tipos elementales de sollicitación:

- Carga puntual, definida por su valor y posición.
- Carga distribuida en línea y superficie, definida por sus componentes según la normal y las tangentes por unidad de longitud o área, o bien por sus componentes según los ejes cartesianos por unidad de longitud o superficie según el plano normal a cada eje.
- Carga gravitatoria, definida por la aceleración de la gravedad y su orientación.
- Carga centrífuga, definida por la velocidad de giro respecto al origen de coordenadas.
- Carga térmica, definida por los incrementos de temperatura

respecto al estado inicial.

El material de cada elemento se considera, en el estado actual del sistema, homogéneo e isótropo, y sus propiedades vienen definidas por el módulo de elasticidad, el módulo de Poisson, el coeficiente de dilatación térmica, la densidad, y el espesor (en tensión plana).

El sistema está dotado actualmente de una rutina de resolución de sistemas de ecuaciones lineales, para sistemas simétricos almacenados en perfil ("sky-line"), capaz de tratar las condiciones de vinculación directamente, dados los desplazamientos prescritos, y de obtener las reacciones correspondientes.

El ensamblaje del sistema de ecuaciones y su resolución, se efectúan mediante un direccionamiento intermedio definible por el usuario del sistema. De esta forma, es posible acoplar al sistema diversos algoritmos de renumeración para minimizar la cantidad de almacenamiento total bajo el perfil, sin alterar el funcionamiento del resto del sistema.

El análisis de sensibilidad puede realizarse integra o parcialmente hasta el segundo orden, según dictamine el módulo de control definido por el usuario del sistema, y está planteado en términos de derivadas direccionales.

El módulo de cálculo y sensibilidad proporciona las derivadas direccionales de primer y segundo orden de los desplazamientos nodales y las tensiones en los puntos de integración respecto a cualquier dirección en el espacio de

diseño, a partir de las derivadas direccionales correspondientes de las variables de entrada, esto es:

- Propiedades del material de los elementos.
- Coordenadas nodales.
- Cargas y desplazamientos prescritos.

A las rutinas que realizan el cálculo estructural y el análisis de sensibilidad de primer y segundo orden se puede acceder en diversos niveles, requiriendo que efectúen todas las operaciones para las que han sido diseñadas o tan solo parte de las mismas. Ya que parte de las operaciones necesarias para efectuar el cálculo pueden ser aprovechadas para realizar el análisis de sensibilidad de primer orden, y parte de la operaciones necesarias para ambos pueden aprovecharse en el análisis de segundo orden, el módulo de control puede configurarse de múltiples formas. Es posible realizar cálculo y análisis de sensibilidad conjuntamente (con objeto de reducir el coste computacional), o separadamente. En este último supuesto puede obtenerse la información estrictamente necesaria del cálculo para realizar el análisis de sensibilidad de primer orden, así como la información estrictamente necesaria de éste para realizar el análisis de sensibilidad de segundo orden, obviando la ejecución de operaciones improductivas.

Las variables de diseño pueden afectar tanto a la forma (coordenadas nodales), como a las dimensiones y propiedades (espesor, módulos elásticos, densidad, coeficientes térmicos), como a las condiciones de vinculación (desplazamientos

prescritos) y acciones externas (valor de cargas puntuales, componentes de cargas distribuidas, orientación de la estructura respecto a la aceleración gravitatoria y valor de ésta, velocidad de rotación de la estructura, incrementos térmicos), sin más que proporcionar al módulo de cálculo y sensibilidad las correspondientes derivadas direccionales en las direcciones de modificación del diseño.

Para reducir el coste de computación, el usuario del sistema puede definir en la interface de cálculo un conjunto de rutinas que dictaminen si en una determinada dirección de modificación del diseño (y en particular en la de una variable de diseño) se altera la forma, dimensiones y propiedades, o condiciones de vinculación y cargas exteriores. Este dictamen es transmitido a las rutinas de análisis de sensibilidad, que evitan de esta forma realizar operaciones innecesarias, evaluando exclusivamente los términos no nulos de las derivadas.

Esta posibilidad es particularmente recomendable en el caso de modelos de parametrización de tipo local, en los que las variables de diseño afectan a parte de la estructura pero no a su conjunto, y especialmente en lo relativo a variables de diseño que determinan la forma de la estructura, ya que es el tratamiento de estas variables el que representa el mayor coste en términos de tiempo de cálculo.

La resolución iterativa del problema de programación matemática subyacente se realiza mediante el algoritmo propuesto en el capítulo anterior, en el que se estima la dirección de avance mediante información suministrada por el cálculo de la

estructura y el análisis de sensibilidad de primer orden completo, y se obtiene el factor de avance mediante una búsqueda unidireccional empleando los desarrollos cuadráticos en la dirección previamente obtenida de la función objetivo y de las funciones que definen las restricciones. Los desarrollos cuadráticos se obtienen a partir del análisis de sensibilidad de segundo orden en la dirección adoptada.

El sistema podrá complementarse en el futuro con otros algoritmos de programación matemática (de tipo dual, de programación cuadrática secuencial, etc.) sin alterar las rutinas restantes. Desde el punto de vista del usuario del sistema ello no implica sino aumentar las posibles configuraciones del módulo de control, y la versatilidad del sistema.

En un apartado posterior se ofrece un patrón recomendable para configurar el módulo de control para un problema de optimización concreto. Este patrón se ha seguido en la realización de los ejemplos tercero, cuarto y quinto presentados en el capítulo VII de este estudio, y se estima considerablemente eficiente.

Dentro de este patrón, el usuario del sistema debe elaborar para cada problema concreto sus correspondientes módulos de definición y parametrización, así como las interfaces de definición y cálculo. El patrón presentado permite optimizar estructuras bajo múltiples hipótesis de carga. Una generalización inmediata de este patrón permite optimizar estructuras bajo múltiples teorías de cálculo. En el ejemplo cuarto (Cap. VII) se plantea y resuelve un caso de optimización

multicarga bajo dos teorías de cálculo diferentes.

La interface de cálculo debe incluir un generador automático de malla. En los ejemplos presentados se han realizado generadores específicos para cada caso. El usuario del sistema puede, sin embargo, acoplar al sistema cualquier tipo de generador automático, con tal de que permita obtener las derivadas direccionales de primer y segundo orden de las coordenadas nodales en cualquier dirección de modificación del diseño.

VI.3 ESTRUCTURA DEL SISTEMA

De acuerdo con los principios metodológicos generales expuestos en el capítulo IV, el sistema DAO² permite la generación de programas modulares de diseño estructural, cuya organización interna puede describirse en la forma:

- Módulo de Definición.
- Módulo de Parametrización.
- Módulo de Cálculo.
- Módulo de Decisión.
- Módulo de Postproceso.
- Módulo de Control.

DAO² está formado por las librerías de rutinas siguientes:

- **MODCAL**
librería de cálculo.
- **MODDER**
librería de análisis de sensibilidad en primer orden.
- **MODDER2**
librería de análisis de sensibilidad en segundo orden.
- **MODSYS**
librería de resolución de sistemas de ecuaciones.
- **MODOPT**
librería de métodos de programación matemática.
- **MODPOS**
librería de postproceso.
- **MODSRV**
librería de servicio.

MODCAL es una librería de métodos de cálculo estructural que en la actualidad dispone de rutinas que permiten el cálculo de estructuras bidimensionales (en tensión plana, deformación plana y simetría de revolución) y tridimensionales (en elasticidad tridimensional) en régimen lineal y estático mediante el Método de Elementos Finitos.

MODDER y MODDER2 son las librerías que permiten realizar el análisis de sensibilidad, de acuerdo con los planteamientos descritos en el capítulo anterior, para las teorías de cálculo citadas.

MODSYS es una librería de resolución de sistemas de ecuaciones lineales almacenadas en perfil, capaz de tratar directamente las condiciones de vinculación (si existen) y evaluar las correspondientes restricciones.

MODOPT es una librería de métodos de programación matemática que permiten resolver el problema general de minimización restringida según el algoritmo descrito en el capítulo precedente.

MODPOS es una librería de rutinas gráficas de propósito general para representación de datos y resultados de programas basados en el Método de Elementos Finitos (representación de mallas bi y tridimensionales, tensiones principales en cálculos bidimensionales, deformadas, eliminación de líneas ocultas, dibujo de diagramas, etc.). En los programas de optimización realizados a lo largo de este estudio la toma de decisiones se realiza de forma totalmente automática. Los resultados parciales y finales del proceso de optimización se almacenan en archivos no

formateados codificados de forma estándar. Varios programas de postproceso permiten a posteriori interpretar gráficamente la información almacenada, generando diversas representaciones tales como las presentadas en el capítulo VII. Dado que en el estado de desarrollo actual de DAO², y de las aplicaciones realizadas hasta la fecha, el módulo de postproceso no se encuentra completamente integrado en el esquema del programa de optimización, y que la estructura de las rutinas gráficas depende fuertemente de la configuración y de las posibilidades de los equipos informáticos disponibles, no describiremos en detalle el contenido de esta librería.

MODSRV es una librería de rutinas de servicio para la realización de operaciones sencillas (productos matriciales, inicialización de campos de memoria, etc.)

Los módulos de definición y cálculo de un programa de optimización estructural generado a partir de DAO² deben ser elaborados por el usuario del sistema para cada familia de problemas en particular, dada la fuerte dependencia de estos módulos respecto al planteamiento de los objetivos técnicos del diseño y a la naturaleza del objeto a diseñar.

El módulo de cálculo se construye a partir de las rutinas de las librerías MODCAL, MODDER, MODDER2 y MODSYS.

El módulo de decisión se construye a partir de las rutinas de la librería MODOPT.

El módulo de postproceso se construye a partir de las rutinas de la librería MODPOS.

Dependiendo de la definición de objetivos, la parametrización adoptada, y las teorías de cálculo y los algoritmos de programación matemática empleados, el usuario debe elaborar las interfaces de definición y cálculo.

El módulo de control de un programa generado a partir de DAO² está formado por el programa principal y diversas subrutinas que acceden a las librerías citadas. En el programa principal el usuario del sistema dimensiona la memoria a utilizar y la localización de los archivos de almacenamiento de variables intermedias. Dependiendo de las rutinas de DAO² que requiere el programa, el dimensionamiento de memoria es distinto.

Esta estructuración permite una gran versatilidad en la generación de programas de optimización estructural y en la selección de las teorías de cálculo y de los algoritmos de solución de sistemas de ecuaciones y programación matemática.

En la Figura 6.1 se esquematiza la composición de un programa de optimización desarrollado a partir de DAO².

Evidentemente, dentro de este marco es posible generar programas de optimización estructural de uso inmediato para diversas familias de problemas, donde se encuentren involucradas las mismas teorías de cálculo, y donde la definición de objetivos y la parametrización del diseño se realiza de una forma sistemática.

Como convenio para la denominación en las rutinas de las librerías mencionadas se han adoptado los siguientes prefijos:

Módulo de Definición.-

DEF\$_____ : rutinas de usuario

Módulo de Parametrización.-

PAR\$_____ : rutinas de usuario

Módulo de Cálculo.-

CAL\$_____ : librería MODCAL
DER\$_____ : librería MODDER
DER_2\$_____ : librería MODDER2
SLE\$_____ : librería MODSYS

Módulo de Decisión.-

OPT\$_____ : librería MODOPT

Módulo de Postproceso.-

POS\$_____ : librería MODPOS

Interface de Cálculo.-

INT_CAL\$_____ : rutinas de usuario

Interface de Definición.-

INT_DEF\$_____ : rutinas de usuario

Figura 6.1.- Composición del esquema de optimización desarrollado a partir del sistema propuesto.

- librería MODCAL.... cal\$_____
- librería MODDER.... der\$_____
- librería MODDER2... der_2\$_____
- librería MODSYS.... sle\$_____
- librería MODOPT.... opt\$_____
- librería MODPOS.... pos\$_____
- librería MODSRV.... srv\$_____

Es recomendable que en la denominación de las rutinas elaboradas por el usuario en cada problema de optimización se adopten los siguientes prefijos:

- Módulo de Definición.....def\$_____
- Módulo de Parametrización....par\$_____
- Interface de Cálculo.....int_cal\$_____
- Interface de Definición.....int_def\$_____

VI.4 DESCRIPCION DE RUTINAS DAO²

A continuación se detalla el contenido de las librerías de rutinas citadas.

La Tabla 6.1 muestra las relaciones entre las rutinas DAO² de las diversas librerías.

Tabla 6.1.- Relaciones entre las rutinas de las diversas librerías

Nombre de Rutina	Rutinas a que accede
cal\$cont_f_d	cal\$scal_f_d cal\$ensamb_f cal\$gen_f_d srv\$ini_vector srv\$sel_subvector
cal\$cont_f_pun	cal\$scal_f_pun cal\$ensamb_f cal\$gen_f_pun srv\$ini_vector
cal\$cont_f_vol_elem	cal\$scal_f_tdi cal\$scal_f_vol cal\$ensamb_f cal\$gen_f_cen_elem cal\$gen_f_gra_elem cal\$gen_f_ter_elem srv\$ini_vector srv\$sel_subvector
cal\$cont_rig_elem	cal\$scal_k cal\$ensamb_k cal\$esc_db cal\$gen_rig_elem srv\$sel_subvector
cal\$cont_tens	cal\$scal_d cal\$gen_tens cal\$lee_db srv\$ini_vector srv\$sel_subvector_2 srv\$sum_subvector
cal\$scal_d	srv\$ini_vector
cal\$scal_db	srv\$mult_m_m
cal\$scal_der_cart	srv\$mult_mt_m
cal\$scal_dt_ter	cal\$interpol_ff srv\$mult_m_m
cal\$scal_f_tdi	srv\$mult_mt_m
cal\$scal_f_vol	srv\$mult_m_m
cal\$scal_gauss_cord	cal\$interpol_ff
cal\$scal_jac	srv\$mult_m_mt
cal\$scal_tens_principales	srv\$scal_angulo

Tabla 6.1.- (Cont. 1)

Nombre de Rutina	Rutinas a que accede
cal\$fin_prog	cal\$cierra_archivos
cal\$gen_f_cen_elem	cal\$scal_gauss_cord
cal\$gen_f_d	cal\$scal_cmat_cd cal\$scal_jac cal\$scal_trans_cd cal\$interpol_ff
cal\$gen_f_ter_elem	cal\$scal_dt_ter
cal\$gen_rig_elem	cal\$scal_b cal\$scal_d cal\$scal_db cal\$scal_der_cart cal\$scal_di_jac cal\$scal_jac cal\$interpol_ff
cal\$gen_tens	cal\$scal_dt_ter srv\$mult_m_m cal\$scal_tens_principales
cal\$ini_alm_datos	cal\$abre_archivos cal\$ini_alm
cal\$ini_ff	cal\$scal_ff_isop
cal\$ini_ff_l	cal\$scal_ff_isop
cal\$ini_parametros	cal\$ini_ff cal\$ini_ff_l cal\$ini_pes_int
der\$scal_d	srv\$ini_vector
der\$scal_db	srv\$ini_vector srv\$mult_m_m
der\$scal_dt_ter	cal\$interpol_ff der\$scal_def_ini_ter srv\$mult_m_m
der\$scal_f_tdi	srv\$mult_mt_m
der\$scal_f_vol	srv\$mult_m_m

Tabla 6.1.- (Cont. 2)

Nombre de Rutina	Rutinas a que accede
der\$cont_deriv_direc	srv\$ini_vector
der\$cont_dka	cal\$ensamb_f srv\$ini_vector srv\$sel_subvector_2
der\$cont_f_d	cal\$ensamb_f der\$scal_f_d der\$gen_f_d srv\$ini_vector
der\$cont_f_pun	cal\$ensamb_f der\$scal_f_pun der\$gen_f_pun srv\$ini_vector
der\$cont_f_vol_elem	cal\$ensamb_f der\$scal_f_tdi der\$scal_f_vol der\$gen_f_cen_elem der\$gen_f_gra_elem der\$gen_f_ter_elem srv\$ini_vector
der\$cont_rig_elem	der\$scal_k der\$esc_d_db der\$gen_rig_elem
der\$cont_tens	der\$scal_d der\$gen_tens der\$lee_d_db srv\$sel_subvector_2
der\$fin_prog	der\$cierra_archivos
der\$gen_f_cen_elem	cal\$scal_gauss_cord
der\$gen_f_d	cal\$scal_jac cal\$interpol_ff der\$scal_cmat_cd der\$scal_trans_cd
der\$gen_f_ter_elem	der\$scal_dt_ter
der\$gen_rig_elem	cal\$scal_der_cart cal\$scal_jac cal\$interpol_ff der\$scal_b der\$scal_d der\$scal_db der\$scal_di_jac

Tabla 6.1.- (Cont. 3)

Nombre de Rutina	Rutinas a que accede
der\$gen_tens	der\$cal_dt_ter srv\$mult_m_m der\$cal_tens_principales
der\$ini_alm_datos	der\$abre_archivos
der_2\$cal_d	srv\$ini_vector
der_2\$cal_db	srv\$ini_vector srv\$mult_m_m
der_2\$cal_dt_ter	cal\$interpol_ff srv\$mult_m_m der_2\$cal_def_ini_ter
der_2\$cal_f_tdi	srv\$mult_mt_m
der_2\$cal_f_vol	srv\$mult_m_m
der_2\$cont_d2ka_dkda	cal\$ensamb_f srv\$ini_vector srv\$sel_subvector_2
der_2\$cont_f_d	cal\$ensamb_f der_2\$cal_f_d der_2\$gen_f_d srv\$ini_vector
der_2\$cont_f_pun	cal\$ensamb_f der_2\$cal_f_pun der_2\$gen_f_pun srv\$ini_vector
der_2\$cont_f_vol_elem	cal\$ensamb_f der_2\$cal_f_tdi der_2\$cal_f_vol der_2\$gen_f_cen_elem der_2\$gen_f_gra_elem der_2\$gen_f_ter_elem srv\$ini_vector
der_2\$cont_rig_elem	der\$esc_d_db der_2\$cal_k der_2\$gen_rig_elem
der_2\$cont_tens	der\$lee_d_db der_2\$cal_d der_2\$gen_tens srv\$sel_subvector_2

Tabla 6.1.- (Cont.4)

Nombre de Rutina	Rutinas a que accede
der_2\$fin_prog	der_2\$cierra_archivos
der_2\$gen_f_cen_elem	cal\$scal_gauss_cord
der_2\$gen_f_d	cal\$scal_jac cal\$interpol_ff der2\$scal_cmat_cd der_2\$scal_trans_cd
der_2\$gen_f_ter_elem	der_2\$scal_dt_ter
der_2\$gen_rig_elem	cal\$scal_der_cart cal\$scal_jac cal\$interpol_ff der_2\$scal_b der_2\$scal_d der_2\$scal_db der_2\$scal_di_jac
der_2\$gen_tens	der_2\$scal_dt_ter srv\$mult_m_m der_2\$scal_tens_principales
der_2\$ini_alm_datos	der_2\$abre_archivos
opt\$direccion_entrada	opt\$normaliza_direccion srv\$ini_vector
opt\$linealizacion	srv\$ini_vector
opt\$maximo_descenso	opt\$normaliza_direccion
sle\$descomposicion_ldlt	sle\$error
sle\$sol_sis_amc	sle\$csp_gdl_res sle\$descomposicion_ldlt sle\$imp_vincul_f sle\$imp_vincul_f sle\$sustituciones_ldlt

VI.4.1 Librería MODCAL

La librería MODCAL está estructurada en tres niveles de rutinas que denominaremos:

- . Primer nivel: Rutinas de distribución.
- . Segundo nivel: Rutinas intermedias de cálculo y gestión de memoria.
- . Tercer nivel: Rutinas especializadas de cálculo elemental.

El primer nivel está formado por rutinas directamente accesibles desde el módulo de control. Estas rutinas realizan funciones superiores (inicialización del programa y direccionamiento de la memoria, obtención y ensamblaje de matrices de rigidez, obtención y ensamblaje de vectores de fuerzas, obtención de tensiones, etc.). Para su acceso desde el módulo de control no es preciso conocer más que en líneas generales el funcionamiento interno de las rutinas y la utilización por las mismas del almacenamiento en memoria. En general el usuario del sistema no necesita acceder directamente a las rutinas de los niveles inferiores.

Las rutinas de distribución se apoyan en las rutinas intermedias de cálculo y gestión de memoria, a las que envían las posiciones de memoria adecuadas para la realización de los cálculos, accediendo a unas u otras según las variables de control de programa (que indican el tipo de análisis a realizar, tipo de elementos, tipo de cargas, etc.). El usuario del sistema puede acceder a este segundo nivel desde el módulo de control, si

bien ello exige un conocimiento más detallado de su funcionamiento que en el acceso a las rutinas del primer nivel.

A su vez, las rutinas de los primeros dos niveles se apoyan en las rutinas de cálculo elemental especializadas en tareas concretas (cálculo de jacobianos, matrices propias del modelo, etc.). El acceso a este nivel desde el módulo de control exige un conocimiento muy detallado de su funcionamiento y especialmente de la forma en que se gestiona el almacenamiento en memoria.

VI.4.1.1 Rutinas de distribución

Las rutinas de distribución se citan a continuación especificando sus correspondientes funciones:

- **cal\$fix_constantes**

definición de las constantes fundamentales del programa (tipo de cálculo, tipos de carga a considerar, etc.)

- **cal\$ini_parametros**

inicialización de parámetros de almacenamiento de memoria, funciones de forma, pesos de integración de las cuadraturas, etc.).

- **cal\$ini_alm_datos**

apertura de archivos e inicialización del almacenamientos matricial (renumeración nodal, punteros de almacenamiento en perfil).

- **cal\$cont_rig_elem**

generación, integración y ensamblaje de matrices de rigidez de elementos.

- **cal\$cont_f_vol_elem**

generación, integración y ensamblaje de vectores de fuerzas volumétricas (gravitatoria, centrífuga, térmica).

- **cal\$cont_f_pun**
generación y ensamblaje de vectores de fuerzas puntuales.
- **cal\$cont_f_d**
generación, integración y ensamblaje de fuerzas distribuidas (dadas las componentes tangenciales y normal al contorno de los elementos, o las componentes en ejes cartesianos).
- **cal\$cont_tens**
obtención de tensiones en puntos de Gauss.
- **cal\$imp_vincul**
tratamiento previo de condiciones de vinculación, inicialización de reacciones.
- **cal\$re_gen_reacciones**
recálculo de reacciones (solo en simetría de revolución, para obtener la reacción por unidad de longitud).
- **cal\$fin_prog**
finalización del programa.

VI.4.1.2 Rutinas intermedias de cálculo y gestión de memoria

Las rutinas intermedias de cálculo y gestión de memoria se citan a continuación especificando sus correspondientes funciones:

- **cal\$gen_rig_elem**
generación de las aportaciones a la matriz de rigidez de elemento de cada punto de Gauss de integración.
- **cal\$gen_f_gra_elem**
generación de las aportaciones al vector de fuerzas de elemento de las cargas gravitatorias en cada punto de Gauss de integración.
- **cal\$gen_f_cen_elem**
generación de las aportaciones al vector de fuerzas de elemento de las cargas centrífugas en cada punto de Gauss de integración.

- **cal\$gen_f_pun**
generación de las aportaciones al vector de fuerzas de elemento de las cargas puntuales en cada punto de Gauss de integración.
- **cal\$gen_f_d**
generación de las aportaciones al vector de fuerzas de elemento de las cargas distribuidas en cada punto de Gauss de integración.
- **cal\$gen_f_ter_elem**
generación de las aportaciones al vector de fuerzas de elemento de las cargas térmicas en cada punto de Gauss de integración.
- **cal\$gen_tens**
generación de las tensiones en puntos de Gauss de integración.
- **cal\$ensamb_k**
ensamblaje de la matriz de rigidez de elemento en la matriz de rigidez global.
- **cal\$ensamb_f**
ensamblaje del vector de fuerzas de elemento debido a un caso de carga tipo en el vector de fuerzas global correspondiente a cada hipótesis de carga.
- **cal\$abre_archivos**
apertura de archivos de almacenamiento intermedio.
- **cal\$cierra_archivos**
cierre de archivos de almacenamiento intermedio.
- **cal\$esc_db**
escritura de matrices " $\tilde{S}=\tilde{DB}$ " de los puntos de Gauss de cada elemento.
- **cal\$lee_db**
lectura de matrices " $\tilde{S}=\tilde{DB}$ " de los puntos de Gauss de cada elemento.

VI.4.1.3 Rutinas especializadas de cálculo elemental

Las rutinas especializadas de cálculo elemental se citan a continuación especificando sus correspondientes funciones:

- **cal\$ini_anc**
inicialización del almacenamiento matricial en perfil, con reenumeración opcional de nodos.
- **cal\$cal_ff_isop**
definición de funciones de forma de la biblioteca de elementos.
- **cal\$ini_ff**
inicialización de funciones de forma de elementos.
- **cal\$ini_ff_1**
inicialización de funciones de forma de contorno de elementos.
- **cal\$ini_pes_int**
inicialización de pesos de integración.
- **cal\$interpol_ff**
interpolación mediante funciones de forma.
- **cal\$cal_k**
integración de la matriz de rigidez de elemento.
- **cal\$cal_f_vol**
integración del vector de fuerzas volumétricas.
- **cal\$cal_f_pun**
cálculo del vector de fuerzas puntuales.
- **cal\$cal_f_d**
integración del vector de fuerzas distribuidas.
- **cal\$cal_f_tdi**
integración del vector de fuerzas debidas a tensiones y deformaciones iniciales (incluidas térmicas).

- **cal\$cal_b**
cálculo de la matriz " \tilde{B} " en puntos de Gauss de integración.
- **cal\$cal_d**
cálculo de la matriz " \tilde{D} " del material.
- **cal\$cal_db**
producto de las matrices " $\tilde{D}\tilde{B}$ ".
- **cal\$cal_trans_cd**
transformación de cargas distribuidas a ejes globales.
- **cal\$cal_cmat_cd**
cálculo de matrices de transformación para cargas distribuidas.
- **cal\$cal_dt_ter**
cálculo de tensiones y deformaciones térmicas en puntos de Gauss.
- **cal\$cal_def_ini_ter**
cálculo de deformaciones iniciales térmicas.
- **cal\$cal_der_cart**
cálculo de derivadas cartesianas de funciones de forma.
- **cal\$cal_jac**
cálculo del jacobiano de la transformación isoparamétrica.
- **cal\$cal_di_jac**
cálculo del determinante y la inversa de la matriz jacobiano.
- **cal\$cal_gauss_cord**
cálculo de coordenadas de puntos de Gauss de integración.
- **cal\$cal_tens_principales**
cálculo de tensiones principales.

VI.4.2 Librería MODDER

La librería MODDER realiza las operaciones de análisis de sensibilidad de primer orden relacionadas con la ecuación de estado del Método de Elementos Finitos.

La librería está estructurada de forma idéntica a la anterior, a la que complementa, y contiene fundamentalmente una rutina homónima de análisis de sensibilidad por cada rutina de cálculo de MODCAL. Algunas rutinas de MODCAL pueden realizar directamente las operaciones correspondientes de análisis de sensibilidad, motivo por el que carecen de complemento en MODDER. Existen en MODDER dos rutinas sin equivalente en MODCAL, cuyas funciones se especificarán. En los otros casos, se asume que las funciones de la subrutina son las de derivación de primer orden de los cálculos efectuados en su homónima de MODCAL.

VI.4.2.1 Rutinas de distribución

Las rutinas de distribución se citan a continuación:

- **der\$cont_deriv_direc**

obtención de la derivada direccional de una función a partir de su gradiente y de la dirección de derivación.

- **der\$cont_dka**

cálculo de los productos matriciales de las derivadas de las matrices de rigidez de elemento por los desplazamientos nodales, para el análisis de sensibilidad de primer orden.

- **der\$ini_alm_datos**

- **der\$cont_rig_elem**

- **der\$cont_f_vol_elem**

- **der\$cont_f_pun**

- der\$cont_f_d
- der\$cont_tens
- der\$re_gen_reacciones
- der\$fin_prog

VI.4.2.2 Rutinas intermedias de cálculo y gestión de memoria

Las rutinas intermedias de cálculo y gestión de memoria se citan a continuación:

- der\$gen_rig_elem
- der\$gen_f_gra_elem
- der\$gen_f_cen_elem
- der\$gen_f_pun
- der\$gen_f_d
- der\$gen_f_ter_elem
- der\$gen_tens
- der\$abre_archivos
- der\$cierra_archivos
- der\$esc_d_db
- der\$lee_d_db

VI.4.2.3 Rutinas especializadas de cálculo elemental

Las rutinas especializadas de cálculo elemental se citan a continuación :

- der\$cal_k
- der\$cal_f_vol
- der\$cal_f_pun
- der\$cal_f_d

- der\$cal_f_tdi
- der\$cal_b
- der\$cal_d
- der\$cal_db
- der\$cal_trans_cd
- der\$cal_cmat_cd
- der\$cal_dt_ter
- der\$cal_def_ini_ter
- der\$cal_di_jac
- der\$cal_tens_principales

VI.4.3 Librería MODDER2

La librería MODDER2 realiza las operaciones de análisis de sensibilidad de segundo orden relacionadas con la ecuación de estado del Método de Elementos Finitos.

La librería está estructurada de forma idéntica a las anteriores, a las que complementa, y contiene fundamentalmente una rutina homónima de análisis de sensibilidad de segundo orden por cada rutina de cálculo de MODCAL y de sensibilidad de primer orden de MODDER. Algunas rutinas de MODCAL o MODDER pueden realizar directamente las operaciones correspondientes de análisis de sensibilidad de segundo orden, motivo por el carecen de complemento en MODDER2. Existe en MODDER2 una rutina sin equivalente en MODCAL, cuya función se especificará. En los otros casos, se asume que las funciones de la subrutina son las de derivación de segundo orden de los cálculos efectuados en su homónima de MODCAL.

VI.4.3.1 Rutinas de distribución

Las rutinas de distribución se citan a continuación:

- **der_2\$cont_d2ka_dkda**

cálculo de los productos matriciales de las derivadas segundas de las matrices de rigidez del elemento por los desplazamientos nodales, y de las derivadas de las matrices de rigidez por las derivadas de los desplazamientos nodales, para el análisis de sensibilidad de segundo orden.

- **der_2\$ini_alm_datos**

- **der_2\$cont_rig_elem**

- **der_2\$cont_f_vol_elem**

- **der_2\$cont_f_pun**

- **der_2\$cont_f_d**

- **der_2\$cont_tens**

- **der_2\$re_gen_reacciones**

- **der_2\$fin_prog**

VI.4.3.2 Rutinas intermedias de cálculo y gestión de memoria

Las rutinas intermedias de cálculo y gestión de memoria se citan a continuación:

- **der_2\$gen_rig_elem**

- **der_2\$gen_f_gra_elem**

- **der_2\$gen_f_cen_elem**

- **der_2\$gen_f_pun**

- **der_2\$gen_f_d**

- **der_2\$gen_f_ter_elem**

- **der_2\$gen_tens**

- **der_2\$abre_archivos**

- **der_2\$cierra_archivos**

- der\$esc_d_db
- der\$lee_d_db

VI.4.3.3 Rutinas especializadas de cálculo elemental

Las rutinas especializadas de cálculo elemental se citan a continuación:

- der_2\$scal_k
- der_2\$scal_f_vol
- der_2\$scal_f_pun
- der_2\$scal_f_d
- der_2\$scal_f_tdi
- der_2\$scal_b
- der_2\$scal_d
- der_2\$scal_db
- der_2\$scal_trans_cd
- der_2\$scal_cmat_cd
- der_2\$scal_dt_ter
- der_2\$scal_def_ini_ter
- der_2\$scal_di_jac
- der_2\$scal_tens_principales

VI.4.4 Librería MODOPT

La librería MODOPT realiza la toma de decisiones en el proceso de optimización.

La librería está concebida como un conjunto de rutinas que permiten resolver un problema general de minimización con

restricciones de igualdad, desigualdad, y laterales.

Consta de las rutinas que se citan a continuación:

- **opt\$chequea_convergencia**

evalúa un criterio de convergencia y dictamina sobre la detención del proceso.

- **opt\$analisis_factibilidad**

realiza un análisis previo a la aplicación del algoritmo de programación matemática, dictaminando si el diseño es factible o no, y clasificando las restricciones según se verifiquen o no, y en que grado se incumplen de acuerdo con unas tolerancias predefinidas.

- **opt\$maximo_descenso**

selecciona como dirección de avance la opuesta al gradiente de la función objetivo si no hay restricciones violadas ni activas.

- **opt\$linealizacion**

formula un problema de programación lineal considerando las restricciones violadas, activas, y aquellas que no se verifiquen suficientemente, según las tolerancias predefinidas. Define las restricciones laterales a tener en cuenta a partir de las restricciones laterales (si existen) del problema de programación no lineal y de los límites móviles impuestos. Realiza una traslación de ejes adecuada para asegurar la no negatividad de las variables primales.

- **opt\$programacion_lineal**

resuelve el problema de programación lineal mediante el Método del Simplex.

- **opt\$selecciona_direccion**

selecciona la dirección de avance a partir de la solución del problema de programación lineal.

- **opt\$normaliza_direccion**

normaliza la dirección de avance.

- **opt\$direccion_entrada**

obtiene una dirección de entrada en la región factible supuestamente eficiente, en el caso de que el problema linealizado carezca de solución o el diseño sea altamente no factible. Emplea el algoritmo detallado en el capítulo anterior.

- **opt\$busqueda**

realiza una búsqueda unidireccional para obtener el factor de avance, tratando de minimizar en una dirección una función cuadrática con restricciones cuadráticas y laterales. Si el problema no tiene solución, adopta un factor de avance tal que la media cuadrática de las violaciones de las restricciones sea mínima. Emplea el algoritmo detallado en el capítulo anterior.

- **opt\$actualizar**

actualiza el valor de las variables de diseño, dadas la dirección y el factor de avance.

- **opt\$comentario**

permite enviar un comentario a una unidad lógica de escritura sobre la evolución del proceso de optimización.

VI.4.5 Librería MODSYS

La librería MODSYS permite resolver sistemas de ecuaciones lineales con prescripciones adicionales del valor de las incógnitas, y matriz de coeficientes simétrica y definida positiva almacenada en perfil mediante el método generalizado de Crout (descomposición LDLt).

La librería permite la solución de un número indefinido de sistemas de ecuaciones (con idéntica matriz) simultáneamente, o a posteriori, una vez efectuada la factorización de Crout. En sucesivas soluciones los valores prescritos para las incógnitas pueden modificarse, pero no el conjunto de incógnitas prescritas.

Consta de las rutinas que se citan a continuación:

- **sle\$sol_sis_anc**

rutina de entrada (llama a las restantes rutinas de la librería)

- **sle\$csp_gdl_res**
rutina de tratamiento previo de las prescripciones adicionales (si existen).
- **sle\$descomposicion_ldlt**
rutina de descomposición generalizada de Crout de la matriz de coeficientes.
- **sle\$imp_vincul_f**
rutina de inicialización de los vectores de incógnitas (si existen prescripciones adicionales).
- **sle\$sustituciones_ldlt**
rutina de sustituciones hacia atrás y hacia delante, que resuelve el sistema de ecuaciones una vez factorizada la matriz.
- **sle\$reacciones**
rutina de evaluación de las reacciones debidas a las incógnitas prescritas.
- **sle\$error**
rutina de control de errores en la resolución del sistema de ecuaciones (matriz no regular o menores principales no regulares)

VI.4.6 Librería MODSRV

La librería MODSRV contiene rutinas generales de servicio que permiten realizar de forma eficiente diversas operaciones sencillas.

Consta de las rutinas que se citan a continuación:

- **srv\$ini_vector**
rutina de inicialización de campos de memoria.
- **srv\$mult_mt_m**
rutina de multiplicación de una matriz rectangular traspuesta por otra matriz rectangular.

- **srv\$mult_m_m**
rutina de multiplicación de dos matrices rectangulares.
- **srv\$mult_m_mt**
rutina de multiplicación de una matriz rectangular por otra matriz rectangular traspuesta.
- **srv\$sel_subvector**
rutina de selección de una parte de los componentes de un vector a través de un puntero de direccionamiento intermedio.
- **srv\$sel_subvector_2**
rutina de selección de una parte de los componentes de un vector a través de dos punteros de direccionamientos intermedios.
- **srv\$sum_subvector**
rutina de adición de una parte de los componentes de un vector a otro vector a través de un puntero de direccionamiento intermedio.
- **srv\$sum_subvector_2**
rutina de adición de una parte de los componentes de un vector a otro vector a través de dos punteros de direccionamiento intermedios.
- **srv\$scal_angulo**
rutina que evalúa un ángulo entre 0 y 2π , conocidos su seno y su coseno.

**VI.5 ESQUEMA PATRON DE OPTIMIZACION ESTRUCTURAL EN EL
SISTEMA DAO²**

El esquema propuesto a continuación puede considerarse como un patrón de relativa generalidad para el desarrollo de programas de optimización estructural a partir del sistema DAO², empleando el algoritmo de programación matemática descrito en el capítulo anterior.

Este esquema es una transcripción abreviada del módulo de control empleado en la realización de los ejemplos tercero, cuarto y quinto que se presentan en el Capítulo VII.

En términos algorítmicos la estructura del programa puede describirse resumidamente como:

```
{ MODULO DE CONTROL }
INICIALIZAR PROGRAMA
mientras no haya convergencia, hacer

    CALCULO Y SENSITIVIDAD DE PRIMER ORDEN
    OBTENCION DE DIRECCION DE AVANCE
    SENSIBILIDAD DE SEGUNDO ORDEN
    OBTENCION DE FACTOR DE AVANCE
    Y ACTUALIZACION DEL DISEÑO
    ESCRITURA DE RESULTADOS PARCIALES
    ANALIZAR CONVERGENCIA

en caso contrario

    ESCRITURA DE RESULTADOS FINALES
    FINALIZAR EJECUCION
```

A continuación se desarrolla este esquema indicando las llamadas a las diversas rutinas del sistema DAO². Las rutinas que comiencen por los prefijos "def\$", "int_def\$", "par\$" e "int_cal\$" deben ser elaboradas por el usuario del sistema para

cada familia de problemas.

Las partes del esquema que se escriben en negrita son rutinas que debe definir el usuario, y las que a su vez están marcadas con flechas son partes del programa de control para cuya realización el usuario goza de casi total libertad (entradas de datos, salidas de resultados, etc.).

VI.5.1 Esquema genérico

```

{ MODULO DE CONTROL }
{=====}
{===== INICIALIZACION DE PROGRAMA }
{=====}

-> apertura de archivos de usuario
-> lectura de datos generales y de control
-> lectura de datos fundamentales de optimización
-> definición de hipótesis de solicitud
-> lectura de diseño inicial

acceder a: PAR$INI_DISEÑO      !- inicialización de
                               ! modelo de diseño
acceder a: PAR$INI_PARAM      !- inicialización de
                               ! modelo de
                               ! parametrización
acceder a: INT_CAL$INI_CAL     !- datos de
                               ! inicialización del
                               ! modelo de cálculo

acceder a: CAL$FIX_CONSTANTES !- definición de tipo de cálculo
acceder a: CAL$INI_PARAMETROS !- inicialización MEF
acceder a: CAL$INI_ALM_DATOS  !- inicialización almacenamiento
acceder a: DER$INI_ALM_DATOS  !-      "      "
acceder a: DER_2$INI_ALM_DATOS !-      "      "

{_____ comienzo de iteraciones}

INICIO_BUCLE_PRINCIPAL:

{=====}
{=====CALCULO Y SENSIBILIDAD DE PRIMER ORDEN}
{=====}

acceder a: SRV$INI_VECTOR      !
    "      "      "          !- inicializaciones de memoria
    "      "      "          !

acceder a: PAR$DEF_DISEÑO      !- definición de propiedades
                               ! fundamentales y ambientales
acceder a: PAR$DER_DISEÑO      !- derivación de propiedades
                               ! fundamentales y ambientales
acceder a: INT_DEF$PROPS       !- evaluación de
                               ! variables de control en
                               ! función de propiedades
acceder a: INT_DEF$DER_PROPS   !- derivación de
                               ! variables de control en
                               ! función de propiedades
acceder a: INT_CAL$DEF_ENTRADA !- definición de datos de
                               ! entrada al módulo de
                               ! cálculo

```

{ _____ FORMACION Y DERIVACION DE LA ECUACION DE ESTADO}

{ _____ cargas no-volumétricas}

bucle sobre tipos de carga

si es carga puntual acceder a: CAL\$CONT_F_PUN

si es carga distribuida acceder a: CAL\$CONT_F_D

bucle sobre variables de diseño

acceder a: INT_CAL\$OPT_CODIGOS

acceder a: INT_CAL\$PREPARA_DER_F_NV

si es carga puntual acceder a: DER\$CONT_F_PUN

si es carga distribuida acceder a: DER\$CONT_F_D

fin de bucle sobre variables de diseño

fin de bucle sobre tipos de carga

{ _____ matriz de rigidez y cargas volumétricas}

bucle sobre elementos

acceder a: CAL\$CONT_RIG_ELEM !- matriz de rigidez

bucle sobre tipos de carga

acceder a: CAL\$CONT_F_VOL_ELEM !- fuerzas volumétricas

fin de bucle sobre tipos de carga

bucle sobre variables de diseño

acceder a: INT_CAL\$PREPARA_DER_RIG !- preparación de
! la derivación

acceder a: DER\$CONT_RIG_ELEM !- derivación de
! matriz de rigidez

bucle sobre tipos de carga

acceder a: INT_CAL\$PREPARA_DER_F_VOL !- preparación
! de derivación

acceder a: DER\$CONT_F_VOL_ELEM !- derivación de
! fuerzas
! volumétricas

fin de bucle sobre tipos de carga

fin de bucle sobre variables de diseño

fin de bucle sobre elementos

{ _____ OBTENCION DE DESPLAZAMIENTOS}

acceder a: INT_CAL\$COND_VINCUL !- obtención de
! condiciones de vínculo

```

acceder a: CAL$IMP_VINCUL      !- imposición de
                                ! de cond. de vinculo
acceder a: SLE$SOL_SIS_AMC    !- primera sol. del sistema
acceder a: CAL$RE_GEN_REACCIONES !- recálculo de reacciones

acceder a: INT_DEF$DESP      !- evaluación de
                                ! variables de control en
                                ! función de desplazamientos
{_____OBTENCION DE DERIVADAS DE DESPLAZAMIENTOS}

bucle sobre elementos
  bucle sobre variables de diseño

    acceder a: INT_CAL$OPT_CODIGOS

      acceder a: DER$CONT_DKA    !- operaciones matriciales
      fin de bucle sobre variables de diseño
    fin de bucle sobre elementos

    acceder a: INT_CAL$DER_COND_VINCUL !- derivación de
                                          ! cond. de vinculo

    acceder a: SLE$SOL_SIS_AMC    !- segunda solución del sistema
    acceder a: INT_DEF$DER_DESP    !- evaluación de derivadas
                                          ! de variables de control en
                                          ! función de desplazamientos
{_____OBTENCION DE TENSIONES Y DERIVADAS}

bucle sobre hipótesis de carga
  bucle sobre elementos

    acceder a: CAL$CONT_TENS      !- cálculo de tensiones

    acceder a: INT_DEF$TENS      !- evaluación de
                                  ! variables de control en
                                  ! función de tensiones

  bucle sobre variables de diseño

    acceder a: INT_CAL$PREPARA_DER_TENS !- prepara la
                                          ! derivación de
                                          ! tensiones

    acceder a: DER$CONT_TENS      !- deriv. de tensiones

    acceder a: INT_DEF$DER_TENS    !- evaluación de
                                  ! deriv. de variables
                                  ! de control en
                                  ! función de tensiones

    fin de bucle sobre variables de diseño
  fin de bucle sobre elementos
fin de bucle sobre hipótesis de carga

```

```

{=====}
{=====OBTENCION DE DIRECCION DE AVANCE}
{=====}

```

```

acceder a: DEF$FOB_REST          !- obtención de
                                ! función obj. y
                                ! restricciones
acceder a: DEF$DER_FOB_REST      !- obtención de derivadas
                                ! de función obj. y
                                ! restricciones

acceder a: OPT$ANALISIS_FACTIBILIDAD !- análisis del diseño

si es factible y no hay restricciones activas,
acceder a: OPT$MAXIMO_DESCENSO !- método del gradiente
si el gradiente es nulo
    ir a FIN_DE_PROCESO (EXITO)
en caso contrario continuar

en caso contrario
acceder a: OPT$LINEALIZACION      !- generar problema
                                ! linealizado
acceder a: OPT$PROGRAMACION_LINEAL !- resolver problema
                                ! linealizado

si el problema linealizado tiene solución
si la solución es el diseño actual
    ir a FIN_DE_PROCESO (EXITO)
en caso contrario
    acceder a: OPT$SELECCIONA_DIRECCION !- obt. dirección
    acceder a: OPT$NORMALIZA_DIRECCION !- normalizar
    continuar
en caso contrario
    acceder a: OPT$DIRECCION_ENTRADA !- direc. de entrada
                                ! en la región fact.

continuar
continuar

```

```

{=====}
{=====SENSIBILIDAD DE SEGUNDO ORDEN}
{=====}

```

```

acceder a: DER$CONT_DERIV_DIREC !- obtención de derivadas
" " " ! direccionales a partir
" " " ! de los gradientes
" " " ! anteriores

acceder a: PAR$DER2_DISEÑO !- derivación segunda
! de propiedades
! fundamentales y ambientales
acceder a: INT_DEF$DER2_PROPS !- derivación segunda de
! variables de control en
! función de propiedades

```


{ _____ OBTENCION DE DERIVADAS SEGUNDAS DE TENSIONES }

bucle sobre hipótesis de carga
bucle sobre elementos

acceder a: INT_CAL\$PREPARA_DER_TENS !- derivadas
acceder a: CAL\$CONT_TENS ! segundas
acceder a: DER\$CONT_TENS ! de
acceder a: DER_2\$CONT_TENS ! tensiones

acceder a: INT_DEF\$DER2_TENS !- evaluación de
 ! derivadas segundas
 ! de variables de
 ! control en función
 ! de desplazamientos

fin de bucle sobre elementos
fin de bucle sobre hipótesis de carga

{=====}
{=====OBTENCION DEL FACTOR DE AVANCE}
{=====}

acceder a: DEF\$DER2_FOB_REST !- obtención de derivadas
 ! segundas de función obj. y
 ! restricciones

acceder a: OPT\$BUSQUEDA !- minimización unidireccional

si el factor de avance es nulo
 ir a FIN_DE_PROCESO (EXITO)

en caso contrario

 acceder a: OPT\$ACTUALIZAR !- actualiza el diseño
continuar

acceder a: OPT\$CHEQUEA_CONVERGENCIA !- analiza convergencia

si ha convergido

 ir a FIN_DE_PROCESO (EXITO)

o si ha superado número máximo de iteraciones

 ir a FIN_DE_PROCESO (FRACASO)

en caso contrario

-> escribir resultados parciales del proceso de optimización
 ir a INICIO_BUCLE_PRINCIPAL

continuar

```
{=====}  
{=====FINALIZACION DE PROGRAMA}  
{=====}
```

FIN_DE_PROCESO (CONDICION):

-> análisis de la condición de detención del proceso

```
acceder a: CAL$FIN_PROG    !- finalización de cálculo  
acceder a: DER$FIN_PROG    !- finalización de derivación  
acceder a: DER_2$FIN_PROG !- finalización de derivación 2ª
```

-> escritura de resultados finales

-> cierre de archivos de usuario

FIN DE PROGRAMA:

VI.5.2 Rutinas elaboradas por el usuario

Describiremos sucintamente a continuación las diversas rutinas que se han citado en el esquema anterior y que deben ser elaboradas por el usuario del sistema para cada problema o familia de problemas. Evidentemente, el usuario del sistema dispone de un amplio margen de maniobrabilidad para elaborar este conjunto de rutinas de los módulos de parametrización y definición, y de las interfaces de definición y cálculo.

- **par\$ini_diseño**

rutina de inicialización de constantes y variables de diseño.

- **par\$ini_param**

rutina de inicialización del módulo de parametrización.

- **par\$def_diseño**

rutina de definición de propiedades fundamentales y ambientales a partir de variables y constantes de diseño.

- **par\$der_diseño**

rutina de derivación de propiedades fundamentales y ambientales.

- **par\$der2_diseño**

rutina de derivación segunda de propiedades fundamentales y ambientales.

- **def\$fob_rest**

rutina de definición de función objetivo y restricciones en función de variables de control.

- **def\$der_fob_rest**

rutina de derivación de función objetivo y restricciones en función de variables de control.

- **def\$der2_fob_rest**

rutina de derivación segunda de función objetivo y restricciones en función de variables de control.

- **int_def\$props**
rutina de definición de variables de control en función de propiedades fundamentales y ambientales.
- **int_def\$der_props**
rutina de derivación de variables de control en función de propiedades fundamentales y ambientales.
- **int_def\$der2_props**
rutina de derivación segunda de variables de control en función de propiedades fundamentales y ambientales.
- **int_def\$desp**
rutina de definición de variables de control en función de desplazamientos.
- **int_def\$der_desp**
rutina de derivación de variables de control en función de desplazamientos.
- **int_def\$der2_desp**
rutina de derivación segunda de variables de control en función de desplazamientos.
- **int_def\$tens**
rutina de definición de variables de control en función de tensiones.
- **int_def\$der_tens**
rutina de derivación de variables de control en función de tensiones.
- **int_def\$der2_tens**
rutina de derivación segunda de variables de control en función de tensiones.
- **int_cal\$ini_cal**
rutina de generación de variables de inicialización del módulo de cálculo.
- **int_cal\$def_entrada**
rutina de generación de variables de entrada al módulo de cálculo.

- **int_cal\$opt_codigos**
rutina que dictamina si una variable de diseño afecta a propiedades o dimensiones, cargas o forma.
- **int_cal\$prepara_der_f_nv**
rutina de preparación de la derivación de primer o segundo grado de las fuerzas no volumétricas (puntuales y distribuidas), que genera las derivadas de las variables de entrada al módulo de cálculo.
- **int_cal\$prepara_der_rig**
rutina de preparación de la derivación de primer o segundo grado de las matrices de rigidez, que genera las derivadas de las variables de entrada al módulo de cálculo.
- **int_cal\$prepara_der_f_vol**
rutina de preparación de la derivación de primer o segundo grado de las fuerzas volumétricas (gravitatorias, centrífugas, térmicas), que genera las derivadas de las variables de entrada al módulo de cálculo.
- **int_cal\$prepara_der_tens**
rutina de preparación de la derivación de primer o segundo grado de las tensiones, que genera las derivadas de las variables de entrada al módulo de cálculo.
- **int_cal\$cond_vincul**
rutina de generación de los desplazamientos prescritos en nodos coaccionados.
- **int_cal\$der_cond_vincul**
rutina de derivación de los desplazamientos prescritos en nodos coaccionados.
- **int_cal\$der2_cond_vincul**
rutina de derivación segunda de los desplazamientos prescritos en nodos coaccionados.